Amendment to the Claims:

This listing of claims replaces all prior versions, and listings, of claims in the application:

1.   (Currently Amended) A method of forwarding a network packet comprises:

reading a table containing a plurality of flags to determine which of the plurality of flags is set or cleared; ~~and~~

~~performing an operation on the packet to decapsulate or encapsulate the packet in accordance with values of the flags~~

determining if the table returned a decap or encap flag;

adding a decap byte count to a packet start offset and subtracting an encap byte count from the packet start offset if the decap and encap flags are set; and

prepending encap bytes to the packet.

2.   (Currently Amended) The method of claim 1 wherein the table~~s~~ is ~~are~~ populated with forwarding information.

3.   (Currently Amended) The method of claim 1 wherein a ~~the~~ forwarding table structure~~s~~ include~~s~~ a control and management structure including a network stack interface and table managers.

4.   (Currently Amended) The method of claim 3 ~~1~~ wherein the table managers manage routing tables and can include a plurality of tables including a layer 4 connection table, a layer 3

3

destination table, a layer 2 bridge table and a layer 2

connection table.

5.   (Currently Amended) The method of claim 1 wherein the

tables includes a flag to indicates whether the bytes should be

stripped from the packet and a field that indicates the number

of bytes to be stripped.

6.   (Currently Amended) The method of claim 1 wherein the

tables includes a field that specifies decapsulation of header

layers up to a the specified layer.

7.   (Currently Amended) The method of claim 1 wherein the

tables includes a field that specifies an identifier of the

current packet encapsulation type.

8.   (Currently Amended) The method of claim 1 wherein the

tables includes a flag that indicates whether bytes should be

prepended to the packet, a field that specifies the number of

bytes and the bytes to be encapsulated.

9.   (Currently Amended) The method of claim 1 wherein the

tables includes a Next Table Type field which indicates that a

further lookup is required and identifies the table type.

10.   (Canceled).

11.   (Currently Amended) <u>A method for encapsulating /</u>

<u>decapsulating packets comprising:</u>

       <u>receiving a packet;</u>

       <u>reading in a first header of the packet and performing a</u>

<u>layer 2 look-up reading a connection table which returns</u>

<u>parameters;</u>

       <u>determining if the table returned a decap or encap flag;</u>

       ~~The method of claim 10 wherein if the decap and encap flags~~

~~are set,~~

       adding <u>a</u> ~~the~~ decap byte count to a packet start offset and

subtracting <u>an</u> ~~the~~ encap byte count from the packet start offset

<u>if the decap and encap flags are set</u>; and

       prepending ~~the~~ encap bytes to the packet.

12.   (Currently Amended) The method of claim 11<u>1</u>~~0~~ further

comprising:

       determining if there is a next table to examine by looking

at the blank field in the currently read table.

13.   (Original) The method of claim 12 wherein if there is a

next table,

       parsing the next header and fetch and read the next table.

14.   (Original) The method of claim 11 wherein if the decap and

encap flags were not set,

5

determine if the encap flag or the decap flag were set.

15.   (Currently Amended) The method of claim 11 wherein if the encap flag was set,

subtract the encap flag byte count from the start offset and prepend the encap flags ~~bytes~~ to the packet.

16.   (Original) The method of claim 11 wherein if the decap flag was set add a decap byte count to the buffer offset and check the next table.

17.   (Original) The method of claim 11 wherein the packet is comprised of one or more headers followed by a payload, the method further comprises:

copying the payload portion of the packet to a packet buffer.

18.   (Original) The method of claim 17 wherein copying may place the packet at an offset in the buffer to make room for any new header that could be prepended to the packet for packet forwarding.

19.   (Currently Amended) A computer program product residing on a computer readable media for forwarding a network packet comprises instructions to cause a computer to:

6

read a table containing a plurality of flags to determine,

which of the plurality of flags is set or cleared; ~~and~~

~~perform an operation on the packet to decapsulate or~~

~~encapsulate the packet in accordance with values of the flags~~

determine if the table returned a decap or encap flag;

add a decap byte count to a packet start offset and

subtract an encap byte count from the packet start offset if the

decap and encap flags are set; and

prepend encap bytes to the packet.

20.   (Currently Amended) The computer program product of claim

19 wherein the ~~tables~~ is ~~are~~ populated with forwarding

information.

21.   (Currently Amended) The computer program product of claim

19 wherein a ~~the~~ forwarding table structures include a control

and management structure including a network stack interface and

table managers.

22.   (Canceled).

23.   (Currently Amended) A computer program product residing on

a computer readable media for forwarding a network packet

comprises instructions to cause a computer to:

receive a packet;

<u>read in a first header of the packet and perform a layer 2</u>

<u>look-up reading a connection table which return parameters;</u>

<u>determine if the table returned a decap or encap flag;</u>

~~The computer program product of claim 22 wherein if the~~
~~decap and encap flags are set, the computer program executes~~
~~instructions to:~~

add <u>a</u> ~~the~~ decap byte count to a packet start offset and

subtracting <u>an</u> ~~the~~ encap byte count from the packet start offset

<u>if the decap and encap flags are set</u>; and

prepend ~~the~~ encap bytes to the packet.

24.   (Currently Amended) The computer program product of claim

23<u>2</u> further comprising instructions to:

determine if there is a next table to examine by looking at

<u>a</u> ~~the~~ blank field in the currently read table.

25.   (Original) The computer program product of claim 24 wherein

if there is a next table, the computer program executes

instructions to:

parse the next header and fetch and read the next table.

26.   (Currently Amended) The computer program product of claim

23<u>2</u> wherein the packet is comprised of one or more headers

8

followed by a payload, the computer program product further

executes instructions to:

copy the payload portion of the packet to a packet buffer.

27.   (Currently Amended) The computer program product of claim

26 wherein instructions to copy place the packet at an offset in

the buffer to make room for any new header that could be

prepended to the packet for packet forwarding.

28.   (Currently Amended) A processor for processing a network

packet comprises:

a computer storage media storing instructions to cause a

computer to:

read a table containing a plurality of flags to determine,

which of the plurality of flags is set or cleared; and

perform an operation on the packet to decapsulate or

encapsulate the packet in accordance with values of the flags;

determine if the table returned a decap or encap flag;

add a decap byte count to a packet start offset and

subtract an encap byte count from the packet start offset if the

decap and encap flags are set; and

prepend the encap bytes to the packet.

29.   (Original) The processor of claim 28 wherein the table

contains forwarding information.

9

30-31. (Canceled).

32. (New) An apparatus comprising:

means for reading a table containing a plurality of flags to determine which of the plurality of flags is set or cleared;

means for determining if the table returned a decap or encap flag;

means for adding a decap byte count to a packet start offset and for subtracting an encap byte count from the packet start offset if the decap and encap flags are set; and

means for prepending the encap bytes to the packet.

33. (New) The apparatus of claim 32 wherein the table is populated with forwarding information.

34. (New) The apparatus of claim 32 wherein a forwarding table structure includes a control and management structure including a network stack interface and table managers.

35. (New) The apparatus of claim 34 wherein the table managers manage routing tables and can include a plurality of tables including a layer 4 connection table, a layer 3 destination table, a layer 2 bridge table and a layer 2 connection table.

10